Language Modeling With Dynamic Bayesian Networks Using Conversation Types and Part of Speech Information

Yangyang Shi Pascal Wiggers Catholijn M. Jonker

Delft University of Technology, 2628 CD Delft

Abstract

In this paper we investigate whether more accurate modeling of differences in language in different types of conversations, e.g. formal presentations vs. spontaneous conversations can improve the quality of a language model. We also investigate whether the modeling of sentence lengths can improve a language model. A language model is an important component of statistical natural language processing systems, such as automatic speech recognizers and spelling checkers, that judges the plausibility of sentence hypotheses. Standard language modeling approaches rely on statistics over word sequences. Our experiments show that modeling the conversation type and part-of-speech tags sequences improves the language models, while modeling sentence length does not.

1 Introduction

A recurring task in natural language processing is to judge whether a sequence of words constitutes a well-formed sentence in a given language or similarly, which of a number of sentence hypotheses is syntactically and semantically most plausible.

An automatic speech recognizer for example, has to choose among sentence hypotheses based on acoustic evidence, while optical character recognition and handwriting recognition hypothesize sentences based on visual information [5, 10]. In spelling correction one wants to identify misspelled words [7] using the context provided by the sentence and in statistical machine translation the fluency of sentences in the target language is rated [1].

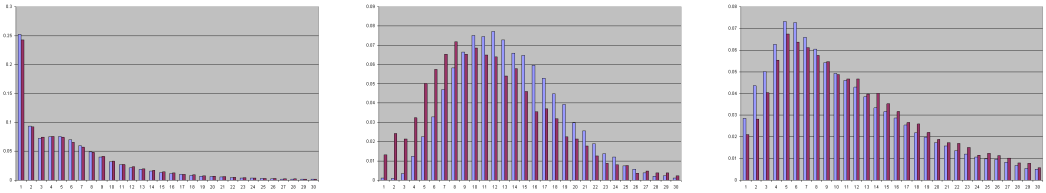
Statistical language models fulfill this task by assigning a probability to every word sequence in a language. The idea is that some word sequences are much more likely than others because of syntactic, semantic and pragmatic constraints. There are multiple ways to define the probability of a sentence, but commonly the chain-rule of probability theory is applied to rephrase the task as assigning a conditional probability to every word in a sentence given the words preceding it:

$$P(w_{1,t}) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2)\dots P(w_n|w_{1,t-1}), \tag{1}$$

where w_i is the i-th word in the sentence and $w_{1,t} = w_1w_2\dots w_t$. From this angle, language modeling can also be seen as predicting the next word in a sentence.

The most common language models, n-grams, are based directly on this idea and predict the next word using a limited history of n preceding words, where n is usually two or three. The advantage of n-gram models is that their parameters can be estimated reliably from a sample corpus. Despite their simplicity these models are surprisingly powerful because their locality makes them robust while at the same time they capture many of the local syntactic and semantic constraints in language.

However, much information that is potentially useful for language modeling gets lost in n-gram models. For example, the assumption that the relative frequency of a word combination is the same for all conversations is clearly incorrect. A word may be more likely in a particular conversation than on average (in a corpus) and far more likely in that conversation than in other conversations.



(a) spontaneous (b) news (c) read

Figure 1: Sentence length distributions for components of the CGN (Northern Dutch)

A second potential problem with the n-gram approach is that, by definition, it prefers short sentences over longer sentences. For spontaneous spoken language this assumption is generally correct but for more formal, written language it does not hold. To illustrate this point, Figure 1 shows the distributions over sentence lengths in different components of the Corpus Spoken Dutch (CGN) [11].

In the research presented in this paper, we investigated whether more accurate modeling of differences in word use among conversation types and of sentence length distributions results can improve language models. We designed six new language models, which we will explain in the section 3. We formulated these models in terms of dynamic Bayesian networks as those provide a natural generalization of statistical language models of n-grams.

Previous research [1, 4, 12] showed that language models can greatly benefit from the inclusion of part-of-speech (POS) information, i.e. information on word categories such as verbs, nouns and adjectives and their relative positions, e.g. the fact that a determiner is often followed by an adjective or a noun. The distributions over POS-tags differ for different conversation types [13], therefore we also include POS tags in some of our models.

We will present our new models in section 3 of this paper. Before that we will provide a brief background on Bayesian networks in the next section. Section 4 discusses the experiments we did to test the performance of the models.

2 Dynamic Bayesian networks

Bayesian networks originate in artificial intelligence as a method for reasoning with uncertainty based on the formal rules of probability theory [9]. A Bayesian network represents the joint probability distribution over a set of random variables $\mathbf{X} = X^1, X^2 \dots X^N$. It consists of two parts:

1. A directed acyclic graph (DAG) G, i.e. a directed graph without any directed cycles. There exists a one to one mapping between the variables X^i in the domain and the nodes v^i of G. The directed arcs in the network represent the direct dependencies between variables. The absence of an arc between two nodes means that the variables corresponding to the nodes do not directly depend on each other.

2. A set of conditional probability distributions (CPDs). With each variable X^i a conditional probability distribution $P(X^i|Pa(X^i))$ is associated, that quantifies how X^i depends on $Pa(X^i)$, the set of variables represented by the parents of node v^i in G representing X^i.

The probabilities are obtained from domain experts, learned from data or a combination of both. Applying the chain rule of probability theory and the independence assumptions made by the network, we can write the joint probability distribution represented by the network in factored form as a product of the local probability distributions:

$$P(X^1, X^2, \dots, X^N) = \prod_{i=1}^{N} P(X^i|Pa(X^i)). \tag{2}$$

Inference in Bayesian networks is the process of calculating the probability of one or more random variables given some evidence, i.e. computing $P(\mathbf{X}^Q|\mathbf{X}^E = \mathbf{x}^E)$ where $\mathbf{X}^Q$ is a set of query variables and $\mathbf{X}^E$ is a set of evidence variables. A number of efficient inference algorithms that exploit the independence of variables in a network exists.

Dynamic Bayesian networks (DBNs) [3, 8] offer a concise way to model processes that evolve over time for which the number of time steps is not known beforehand. A DBN can be defined by two Bayesian

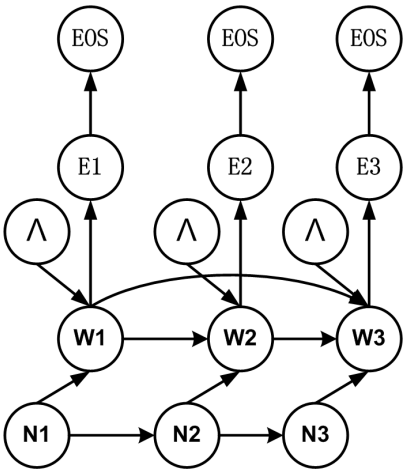


Figure 2: DBN representation of an interpolated trigram (i-trigram)

networks: an a prior model $P(\mathbf{X}_1)$ and a transition model that defines how the variables at a particular time depend on the nodes at the previous time steps:

$$P(\mathbf{X}_t|\mathbf{X}_{t-1}) = \prod_{i=1}^{N} P(X_t^i|Pa(X_t^i)) \tag{3}$$

were $\mathbf{X}_t$ is the set of variables at time t and X_t^i is the ith variable in time step t. The parents of a node can either be in the current or in a previous time slice. Typically, first order Markov assumptions are made, i.e. the nodes in a time slice only depend on the nodes in the previous time slice. The advantage of DBNs for language modeling is that one can define rich models without having to worry about the details of special purpose inference algorithms [12].

3 Models

As our baseline in this research we use an interpolated trigram language model, which is a common choice in language modeling, particularly for speech recognition. This model is based on a trigram, i.e. an n-gram with $n = 3$.

Even though the assumption that a word only depends on the previous two words is a crude approximation of the dependencies in language, it is already difficult to obtain reliable estimates for the parameters of the model. This is especially true since language has a rather skewed distribution. To overcome this problem, language models are typically smoothed. One way to achieve this is by interpolation with lower order n-grams such as bigrams ($n = 2$) and unigrams ($n = 1$):

$$P_{int}(w_i|w_{i-1}w_{i-2}) = \lambda_1 P(w_i|w_{i-1}w_{i-2}) + \lambda_2 P(w_i|w_{i-1}) + \lambda_3 P(w_i), \tag{4}$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Figure 2 shows the DBN representation of the standard interpolated trigram. In every time slice, it consists of a word variable W that depends on the previous two words and on the interpolation variable Λ that takes the three interpolation weights as its values. Its CPD is thus given by equation 3.3. Exceptions are made for the first two words of a sentence. The N variable is a counter that indicates the number of words preceding a word in a sentence. W is conditioned on N such that the current word does not depend on any preceding words if $N = 0$ and if $N = 1$ it only depends on the previous word. The binary E variables model the end of a sentence, while EOS models the end of a complete text. This last node is necessary to ensure that the model is a proper probability model in the sense that the sum of the probabilities over all possible texts is 1.

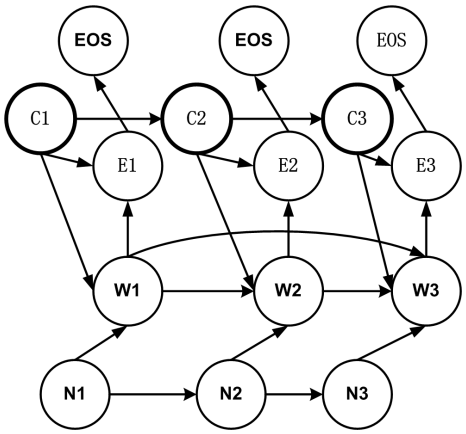


Figure 3: Interpolated trigram with conversation type (i-trigram-c)

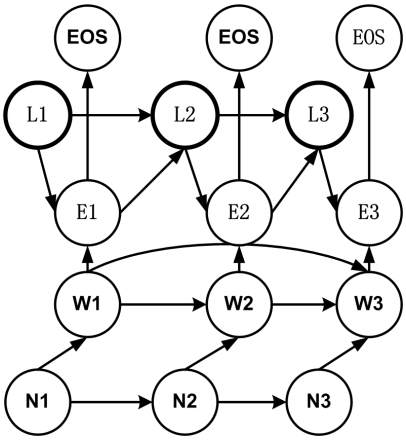


Figure 4: Interpolated trigram with sentence length (i-trigram-l)

3.1 Conversation type

The sentence structure and vocabulary varies greatly with different conversation types. For example, spontaneous speech consist of short sentence with many interjections, adverbs, pronouns and incomplete, while in formal and read speech, longer, grammatically more complicated sentences are used which contain more nouns and determiners [13].

To account for these effects, we added the conversation type as a variable in the model. In Figure 3 , the variable C indicates the conversation type. By conditioning the word variable W on the conversation type, the model can dynamically adapt to the type of conversation. There are many ways in which the influence of the conversation type on the words can be modeled. After some experimenting, we chose to include a term $P(w_i|c_i)$ in the interpolated word distribution:

$$P_{int}(w_i|w_{i-1}w_{i-2}c_i) = \lambda_1 P(w_i|w_{i-1}w_{i-2}) + \lambda_2 P(w_i|w_{i-1}) + \lambda_3 P(w_i) + \lambda_4 P(w_i|c_i). \quad (5)$$

This formulation has the advantage that the probability of a conversation type is not reduced to zero as soon as the model encounters a word-conversation type combination that is not in the training data.

$$\sum_{n=1}^{\infty} \frac{exp^{-6} * 6^n}{n!}$$

3.2 Sentence length

As n-gram language models assign probabilities to sentences by multiplying word probabilities, they typically prefer short sentences over long sentences. As was shown in Figure 1 this fits well with the nature of

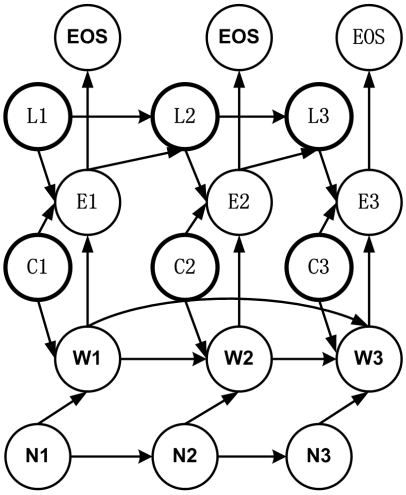


Figure 5: Interpolated trigram with conversation type and sentence length (i-trigram-l-c)

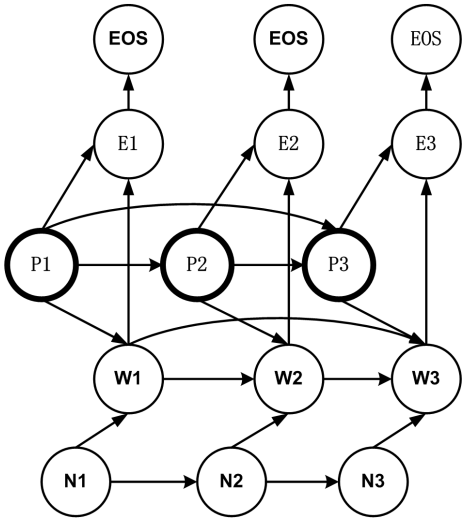


Figure 6: Interpolated trigram with part of speech (i-trigram-p)

spontaneous speech but not with more formal speech such a broadcast news or read speech.

Figure 4 shows an interpolated trigram in which sentence length is explicitly modeled. In this model L is a counter variable that counts the word positions within a sentence. The end of sentence variable E is conditioned on this variable to ensure that the model will learn the distribution over the sentence lengths.

As mentioned above the distribution of sentence lengths varies greatly with different types of conversations. Therefore we created a model in which the sentence length distribution depends on the conversation type (Figure 5).

3.3 Part of speech tags

Including part of speech tags in a language model usually makes the model better [2] as it requires less data to find reliable statistics on the combinations of POS tags that can occur than on the combinations of words. In addition, we can also relate the POS tags to other variables in the model such as the end of the sentence and the conversation type, to make the prediction of the values of those variables more accurate.

Figure 6 shows a model that includes part-of-speech tags. Every part-of-speech depends on the previous two POS-tags, this allows the model to encode simple grammatical constructions. Like the word distribution, the POS tag distribution should be an interpolated distribution to ensure that the model will assign a non-zero probability to every sentence. Every word W is restricted by its own POS tag P. Since the part-of-speech

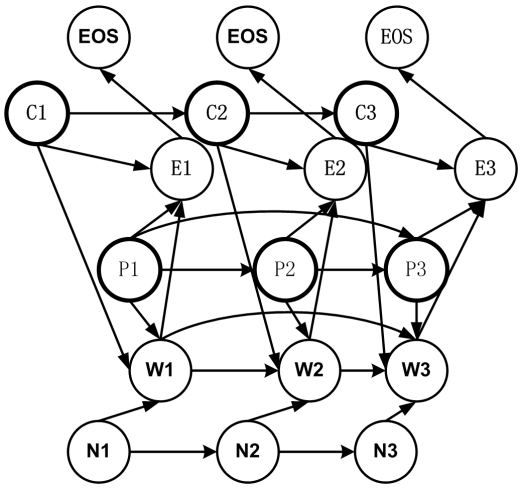


Figure 7: Interpolated trigram with part of speech and conversation type(i-trigram-p-c)

is a property of a word, it is added a conditioning variable in each of the three components of the word distribution (Equation).

We created two other models. One is shown in Figure 7. It depicts a model with conversation types and part of speech. Each of these two factors affect the word respectively. The other model, which combines all three types information: conversation type, part of speech and sentence length. The influence between the three factors themselves and the influence of the three factors to other nodes in the model are the same in the models we discussed before.

4 Experiments

To find out whether our models can improve over standard interpolated trigrams, we trained and tested the models on the Corpus Spoken Dutch (CGN). A corpus of standard Dutch as spoken in the Netherlands and Flanders. The corpus is based on collected audio recordings. For those experiments we used the transcriptions of those audio fragments that are distributed with the corpus. The corpus is subdivided in 15 components that contain different types of speech, ranging from spontaneous conversations to more formal speech. We used the following components:

- *comp-d* spontaneous telephone dialogues,
- *comp-f* broadcasted interviews, discussions and debates
- *comp-h* lessons recorded in a classroom
- *comp-k* broadcasted news
- *comp-l* broadcasted commentaries, columns and reviews
- *comp-m* ceremonial speeches and sermons
- *comp-n* lectures and seminars

The set contains a total of 2843655 words 80% of which we use for training, 10% for development testing and tuning and the remaining 10% for evaluation. We created a vocabulary of 21865 words and 293 different parts-of-speech. The vocabulary contains all unique words that occur more than once in the training data. All words in the data that are not in the vocabulary were replaced by a out-of-vocabulary token. Each of the seven sets listed above represents a conversation type. This implies that a complete document always has a single conversation type.

For training, the values of all variables were provided therefore we calculated all distributions using maximum likelihood estimation. The interpolation weights were trained on the development test set using the expectation-maximization algorithm.

Table 1: *Perplexity results on* CGN *components d,f,h,k,l,m,n*

models	perplexity
i-trigram	340.47
i-trigram-l	344.32
i-trigram-c	312.13
i-trigram-l-c	317.34
i-trigram-p	309.79
i-trigram-p-c	304.37
i-trigram-p-c-l	310.53

We evaluated the models in terms of perplexity, a standard measure in language modeling which is based on the cross-entropy of the language model and a test data set (see for example [6]). The better the model fits the data set, the lower the perplexity will be. Perplexity is calculated as:

$$PP(w_{1,t}) = 2^{-\frac{1}{t}\log P(w_{1,t})}. \tag{6}$$

where $P(w_{1,t})$ is the probability assigned by the model to the test data set. We assume independence of individual documents, therefore the probabilities of the documents are multiplied to obtain this result.

We tested the performance of the all models discussed in the previous section on the held out evaluation set. During this test only the words, the word positions and the punctuation (i.e. sentence ends) were provided to the model, all other variables were treated as hidden variables. Table 1 shows the resulting perplexities.

From these results we can see that explicitly modeling the conversation type does lower perplexity. As should be expected from similar research including POS-tags also improves the model. Combination of those factors results in the lowest perplexity we achieved, a reduction of slightly over 10%. Modeling sentence length on the other hand does not help. In all cases it actually slightly hurt performance.

5 Conclusion

The statistical language models play a important role in natural language processing systems by making a judgement of the probability of sentences. In this paper we presented six new language models which not only focused on the statistics on the word sequences, but also considered the conversation type, part of speech tags, and sentence length which are not used in standard language models. The part of speech provides syntactic information of the speech and the 7 conversation types are forms such as spontaneous conversation and formal presentation. We implemented these new models as Dynamic Bayesian Networks and compared them with the standard trigram language model. The perplexity results of the experiments show that the new language models with conversation type, with part of speech and with both of them improve upon the standard trigram by almost 8%, 9% and 10%, respectively. But the sentence length information did not contribute to improve the trigram language model. In fact, the results are not as good as those of the trigram language models. In the future we plan to investigate whether different interpolation schemes can further improve these models and introduce additional context information such as the topic of a conversation.

References

[1] Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[2] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[3] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150, 1989.

[4] J. Goodman. A bit of progress in language modeling. Technical report, Microsoft Research, 56 Fuchun Peng, 2000.

[5] Jianying Hu, Michael K. Brown, and William Turin. HMM based on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, October 1996.

[6] Daniel Jurafsky and James H. Martin. *Speech and Language Processing - Second Edition*. Prentice Hall, 2009.

[7] Eric Mays, Fred J. Damerau, and Robert L. Mercer. Context based spelling correction. *Information Processing and Management*, 27(5):517–522, 1991.

[8] Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[9] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems - Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

[10] R Plamondon and S.N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:63–84, 2000.

[11] I. Schuurman, M. Schouppe, H. Hoekstra, and T. van der Wouden. CGN, an annotated corpus of spoken Dutch. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, Budapest, Hungary, 14 April 2003.

[12] Pascal Wiggers and Leon J. M. Rothkrantz. Topic-based language modeling with dynamic Bayesian networks. In *the Ninth International Conference on Spoken Language Processing (Interspeech 2006 - ICSLP)*, pages 1866–1869, Pittsburgh, Pennsylvania, September 2006.

[13] Pascal Wiggers and Leon J. M. Rothkrantz. Exploratory analysis of word use and sentence length in the Spoken Dutch Corpus. In Václav Matousek and Pavel Mautner, editors, *Lecture notes in Artificial Intelligence 4629: Text, Speech and Dialogue 2007*, 2007.